NASA CR-130108

STAR

ADAPTION OF A CORRECTOR MODULE
TO THE IMP DYNAMICS PROGRAM

Avco Systems Division
201 Lowell Street
Wilmington, Massachusetts 01887

(NASA-CR-130108) ADAPTION OF A CORRECTOR MODULE TO THE IMP DYNAMICS PROGRAM Final Report and user's manual (Avco Corp., Wilmington, Mass.) Sep. 1972 103 p CSCL 09B G3/08

N73-10238

Unclas
46607

September 1972
Final Report and User's Manual
Contract NAS5-11881
AVSD-0335-72-RR

Prepared for

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND 20771

ADAPTION OF A CORRECTOR MODULE TO THE IMP DYNAMICS PROGRAM

Avco Systems Division
201 Lowell Street
Wilmington, Massachusetts 01887

September 1972
Final Report and User's Manual
Contract NAS5-11881
AVSD-0335-72-RR

Prepared for

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND 20771

TABLE OF CONTENTS

LIST OF FIGURES

1.0 INTRODUCTION

The corrector module of RAEIOS Program (Reference 1) and the IMP Dynamics Computer Program (Reference 2) have been combined to achieve a data fitting capability with the more general spacecraft dynamics models of the IMP Program. The IMP Dynamics Program models spacecraft dynamics for satellites having long flexible booms. The resulting equations of motion are a system of ordinary and partial differential equations. The use of mode shapes to represent the deformations of the flexible booms converts the partial differential equations into ordinary differential equations. The resulting global system of ordinary differential equations is nonlinear, time-dependent in some of its forcing functions and, at least piecewise, time-invariant in its system parameters such as antenna boom stiffness, damping, thermal properties, etc.

In order to start the IMP Dynamics Simulator, initial values of state variables are required. These state variables include central core attitude angles and rates, antenna tip displacements, and velocities, etc. Estimates are also required of the system parameters. The closeness of fit of the Dynamics Simulator depends, among other things, on the accuracy of these estimates.

The general corrector scheme devised to supply "best" estimates of the uncertain system parameters and missing initial conditions is an iterative procedure. The overall logic, shown in Figure 1, is applicable to practically all corrector or optimization schemes. The comparator computes the difference between simulated behavior and measured data in order to generate error signals. The performance criterion combines all error signals

into a single scalar variable which expresses the quality of the fit between simulated and measured data. The Corrector allows a choice of different structures for the performance criterion, i.e., weighted or unweighted and absolute values of errors or squares of errors. The weighted, squared-error performance criterion structure can be thought as (and is in fact equivalent to) the introduction of a diagonalized error covariance matrix, which is often used in filtering problems.

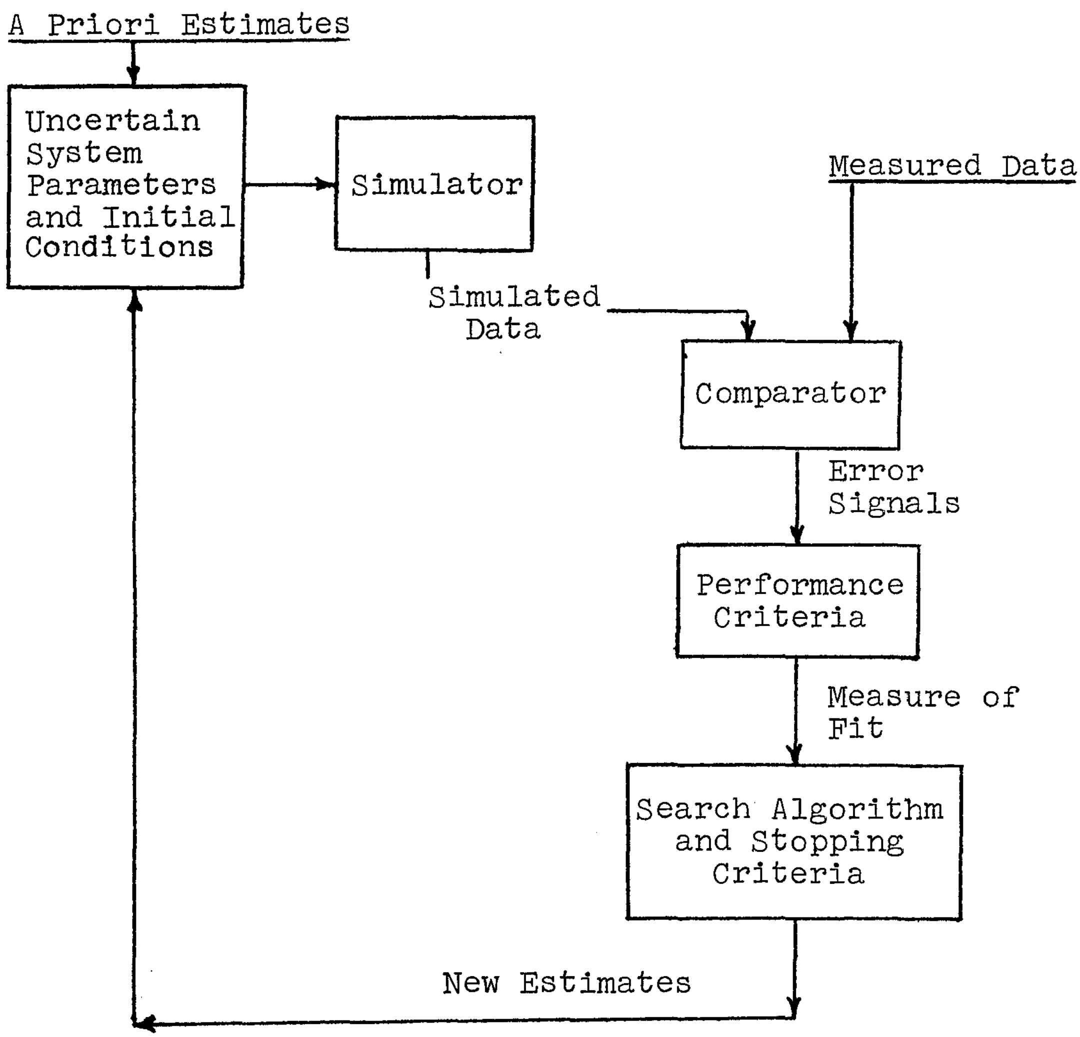


Figure 1 - Corrector Schematic

The remainder of this report describes the properties of the corrector. The technical discussion is concerned with the description of the performance criteria and the search logic for parameter estimation. The final section gives a description of the modifications made to add the corrector to the IMP Dynamics Program. This section includes subroutine descriptions, common definitions, definition of input, and a description of output.

2.0 TECHNICAL DISCUSSION

A variety of search algorithms exist and were examined before selection of the method used in the Corrector. The Rosenbrock rotating coordinate scheme which belongs to the so-called direct search class was selected for the following reasons:

1. Because of the complex nonlinear and time dependent nature of the Dynamics Simulator, forming mathematical expressions for the derivatives of the state equations needed for gradient search methods is a very cumbersome task. Of course, it is possible to estimate numerically such derivatives, but such estimates are very sensitive to noise amplification and, moreover, would require a great number of simulation runs for a high dimensional state-space. Direct methods avoid exact estimates of gradients.
2. Direct search methods, especially of the rotating coordinates variety, are quite independent of the nature of the criterion surface in parameter space, i.e., they are applicable to a great variety of criterion surfaces that do not even have to be continuous. While gradient techniques may be faster in convergence on nearly linear or nearly quadratic portions of surfaces (as may be the case near the optimum), they may be much slower on more irregular surfaces or even diverge completely. The Rosenbrock method, on the other hand, has forced improvements at nearly every step. Because of the complexity of many satellites, it is not, a priori, evident what

are the characteristics of their criterion surfaces. Therefore, the Rosenbrock method, with its wider field of applicability, appeared to be a safer choice.

3. The direct search algorithm adapts itself very easily to changes in the Dynamics Simulator and to changes in search parameters, even if these changes are not trivial.

The Rosenbrock search technique, as described in Reference 3, searches for the greatest or least value of a function of several variables. The method is an algorithm for selecting trial values for the input parameters of a given system model in such a way that some error function of the performance of the system tends to be optimized. The technique has been referred to as the method of "rotating coordinates" because of the manner in which the direction of the variables are perturbed after a series of successful and unsuccessful searches have been made. Rather than varying the input parameters of the system model one at a time, the method rotates the search coordinate system, which initially coincides with the parameter axes in parameter space, so that one search axis points in the most favorable direction as determined from the previous search. The remaining search axes, which are mutually orthogonal, are obtained with a Gram-Schmidt orthogonalization procedure. The search is continued along each of these directions one at a time using the logic which is described below, and then a new set of axes are developed.

The following list of initial conditions and spacecraft properties can be used as parameters in the optimization procedure:

1. Initial conditions on attitude and attitude rates
2. Initial conditions on antenna displacements and velocities
3. Initial conditions on damper angle and angular rate
4. Antenna frame Euler angles
5. Antenna modulus of elasticity
6. Antenna temperature gradients
7. Antenna lengths
8. Antenna no load offsets
9. Antenna damping coefficients
10. Libration damper magnetic hysteresis decay factor
11. Libration damper magnetic hysteresis saturation torque
12. Libration damper magnetic hysteresis initial magnetic moment
13. Libration damper suspension spring constants
14. Libration damper suspension stop angle
15. Libration damper suspension viscous damping coefficient.

2.1 Performance Criterion

An error function or performance criterion is determined from a comparison of actual flight data and simulated data over a specified time span. The flight data is read from a tape and stored in common for use in evaluating the performance criterion. Flight data must be supplied at a constant time interval over the specified time span (for instance, one minute intervals). The flight data is stored in a vector having time and the data for each data channel at every equally spaced time point in the specified data span.

The performance criterion is constructed for each data channel during each trial simulation. At each data time a simulated data vector is formed. An error vector is computed as the difference between the flight data and the simulated data. Provision is made to weight and normalise the error vector to achieve a weighted and normalised error vector if desired, as follows.

$$\delta'_{ij} = \frac{k_i \delta_{ij}}{a_{ij}} \tag{1}$$

Where:

k_i = Weighting factor for the i th data channel.

δ_{ij} = Difference between flight data and simulated data for the i th data channel at the j th time.

a_{ij} = Normalises δ_{ij} to the corresponding flight data. If the flight data is less than the normalisation constant, A_i, δ_{ij} is normalised to A_i. This avoids division by zero.

$$a_{ij} = \begin{cases} a_{ij} & \text{IF } |a_{ij}| \geq A_i \\ A_i & \text{IF } |a_{ij}| < A_i \end{cases} \tag{2}$$

The accumulated error in each data channel is computed by adding either the absolute value or the square of δ_{ij} to the appropriate data channel residual.

$$R_i = \sum_j |\delta'_{ij}| \qquad \text{Absolute Value Criterion}$$

$$R_i = \sum_j (\delta'_{ij})^2 \qquad \text{Squared Criterion} \tag{3}$$

At the end of the specified data span, the total performance criterion is obtained by summing the accumulated residual errors for each data channel.

$$J = \sum_i R_i \tag{4}$$

2.2 Search Logic

Each time the performance criterion is computed constitutes a "trial". A trial is a "Success" if the performance criterion is equal to or less than the value obtained on the previous trial; otherwise it is a "failure". Trials are made along one search axis in parameter space until there is a failure which has been preceded by at least one success. The length of step for these successive trials are determined as follows:

i) An initial step size of e_n is tried. The initial values of e_n in each direction are input to the program; subsequent values are dependent on previous trials. In particular, if d_n is the algebraic sum of all successful trials in the nth direction, then the initial trial of the next search in the nth direction will be γd_n where $0 < \gamma < 1$

ii) After a successful trial, the length of the previous step is multiplied by a constant α $(\alpha > 1)$ and this is added to the previous value used.

iii) If the initial step in a given direction is a failure, then the initial step is replaced by $-\beta e_n$, $(0 < \beta < 1)$ If this fails, the step is multiplied by an additional $-\beta$ etc. This procedure must result in an eventual success since in the limit the search will return to the initial starting point.

After a set of trials has been completed on one search axis, the program searches along the next orthogonal search axis until all search axis have been treated. A new set of search axis is then calculated. All of the trials along the search axis and the

subsequent calculation of a new set of search axes is called a "stage". The number of search axes correspond to the number of parameters being optimized. The rotating search axes are related to the parameter axes by the direction cosine matrix $[C_{\ell,n}]$ with which steps on a given search axis can be resolved into parameter changes. For the first stage $[C_{\ell n}]$ is a unit matrix so that each step, e_n corresponds to a change in only one of the system parameters. For each subsequent stage, a new direction cosine matrix is computed using the Gram-Schmidt procedure as follows:

Let $\hat{\xi}_1^{o}, \hat{\xi}_2^{o} \cdots\cdots \hat{\xi}_N^{o}$ be the set of orthogonal unit vectors defining the directions in the original stage. Suppose that d_1 is the algebraic sum of all successful steps e_1 in the direction $\hat{\xi}_1^{o}$, etc. Then define the set of vectors:

$$
\begin{aligned}
\vec{A}_1 &= d_1 \hat{\xi}_1^{o} + d_2 \hat{\xi}_2^{o} + \cdots\cdots\cdots d_N \hat{\xi}_N^{o} \\
\vec{A}_2 &= \qquad\quad d_2 \hat{\xi}_2^{o} + \cdots\cdots\cdots d_N \hat{\xi}_N^{o} \\
&\vdots \\
\vec{A}_N &= \qquad\qquad\qquad\qquad\qquad\quad d_N \hat{\xi}_N^{o}
\end{aligned}
\tag{5}
$$

The orthogonal unit vectors $\hat{\xi}_1^{\prime}, \hat{\xi}_2^{\prime} \cdots\cdots \hat{\xi}_N^{\prime}$ for the next stage are now obtained using the following vector equations:

$$
\begin{aligned}
\vec{B}_1 &= \vec{A}_1 \\
\hat{\xi}_1^{\prime} &= \vec{B}_1 / |\vec{B}_1| \\
\vec{B}_2 &= \vec{A}_2 - (\vec{A}_2 \odot \hat{\xi}_1^{\prime})\, \hat{\xi}_1^{\prime} \\
\hat{\xi}_2^{\prime} &= \vec{B}_2 / |\vec{B}_2| \\
&\vdots \\
\vec{B}_N &= \vec{A}_N - \sum_{j=1}^{N-1} (\vec{A}_N \odot \hat{\xi}_j^{\prime})\, \hat{\xi}_j^{\prime} \\
\hat{\xi}_N^{\prime} &= \vec{B}_N / |\vec{B}_N|
\end{aligned}
\tag{6}
$$

The new direction cosine matrix is obtained by taking the inverse of the matrix comprised of the components of $\hat{\xi}'_j$ along the parameter axes.

A typical search sequence for a two parameter case is shown in Figure 2 to illustrate the above procedure. Contours correspond to constant values of the performance criterion J. The first two stages shown correspond to 13 trials.

In order to terminate the search, two controllable completion criteria are included. The first completion criterion terminates the optimization procedure if the performance criterion from three successive stages falls within a specified band. The second criterion terminates the procedure if the ratio of three successive performance criteria is less than a specified value. The basic features of the completion criterion are shown in Figure 3. The parameter D which defines the band for the first criterion may vary between zero and one. Specifying a value near 1.0 would emphasize the second criterion which is controlled by the value of the specified ratio, R. A value of D near 0.0 would make the termination dependent upon the magnitudes of successive J's as shown in Figure 3.

Specifying a small value of R near zero will terminate the optimization near a minimum of the performance criteria providing the convergence is not excessively slow. If the convergence is very slow, the first criterion will terminate the problem.

The computer running time for the corrector is greatly dependent upon the input constants used for D and R of the search termination. The particular values used depends on the rate that the problem converges. Initially, the value of D should be small until the convergence rate can be observed. If the convergence rate is rapid, D can be set nearly equal to 1.0. If R $\approx$ 0, the problem will terminate only after finding a minimum if D = 1.0. For initial studies, a value of R between 0 and 0.5 will terminate the search before a minimum was reached and conserve computer running time. The value of R should not be less than or equal to

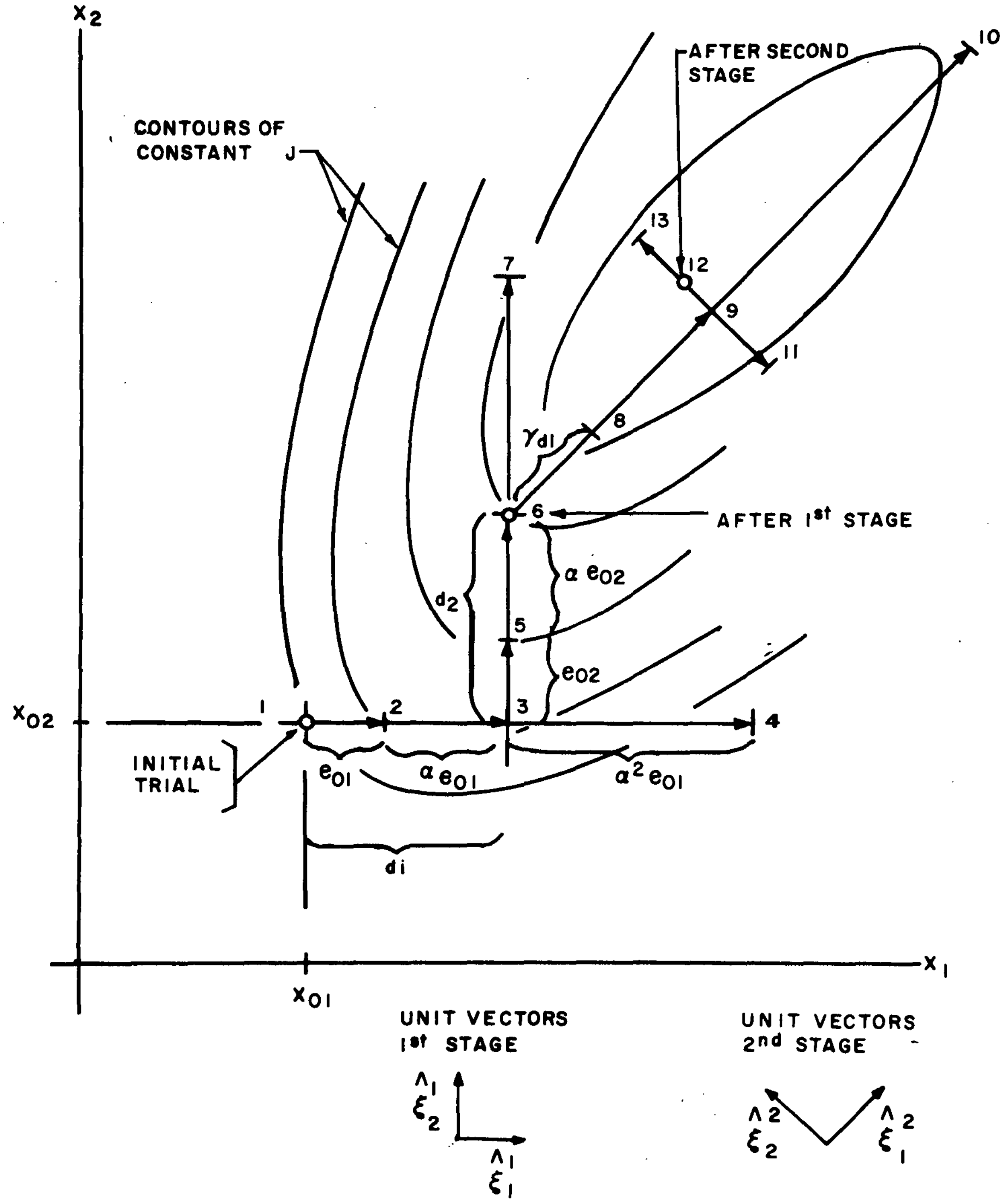
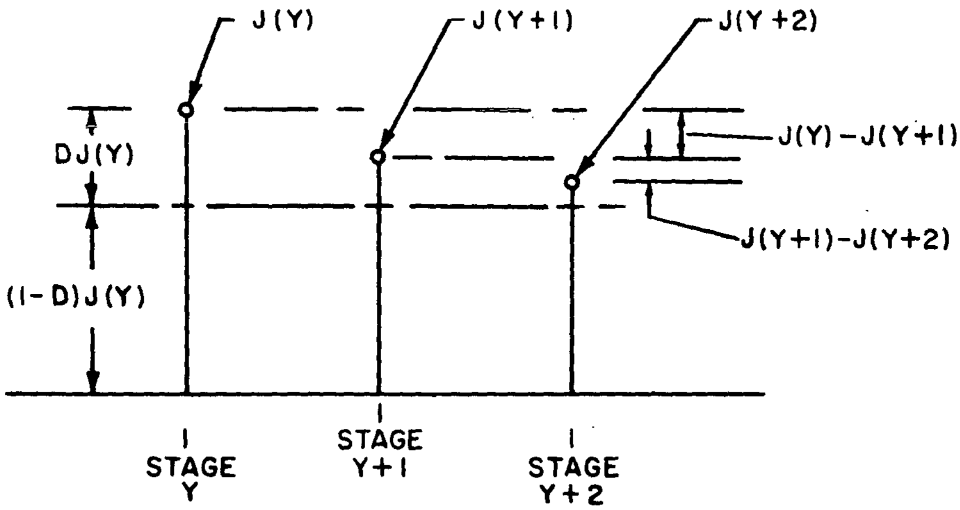


Figure 2 - Typical Search Sequence for Two Parameter Case



SEARCH IS COMPLETED IF BOTH OF THE FOLLOWING CONDITIONS ARE SATISFIED:

CONDITION 1: $(1 - D)\, J(Y) < J(Y + 1)$

$(1 - D)\, J(Y) < J(Y + 2)$

CONDITION 2: $\dfrac{J(Y + 1) - J(Y + 2)}{J(Y) - J(Y + 1)} < R$

Figure 3 - Search Completion Criterion

zero since successive values of the performance criterion, J, are always less than preceeding values.

As a further safeguard, the maximum number of trials conducted in a search is specified through input. Although the search may not be complete upon the execution of the specified trials, an indication of search progress can be obtained from the resultant printout. The user may then alter input and resubmit to obtain optimized parameters. The altered input will reflect the trends and results established by the incomplete search.

The Rosenbrock method for finding the minimum or maximum value of a function suffers from the same limitations of most optimization procedures in that it assumes the function J (y) is unimodal, i.e., has only one minimum over the range of values of interest. In general, the program will converge to a minimum of J. One approach to exploring the uniqueness of this minimum is to initiate the procedure with new initial values of the parameters; the process should always converge to the same optimum parameters if the function J (y) is unimodal.

3.0 PROGRAM DOCUMENTATION

3.1 Subroutine Descriptions

The new subroutines are documented in the following paragraphs. The modifications made to existing routines are described below. The modified flow chart of the main routine is shown in Figure 4.

MAIN	- The main routine was modified to control corrector operation.
CØNVRT	- This subroutine was modified to add a single coefficient for bending stiffness affecting all antennas.
GØUT	- This subroutine was modified to suppress printing of all corrector trials if desired.
GPRINT	- This subroutine was modified to suppress printing of all corrector trials if desired and to write test flight data sets.
SØUT	- This subroutine was modified to suppress printing of all corrector trins if desired.
SØLAR	- Modified to save space
CØMBNZ	- Modified to save space

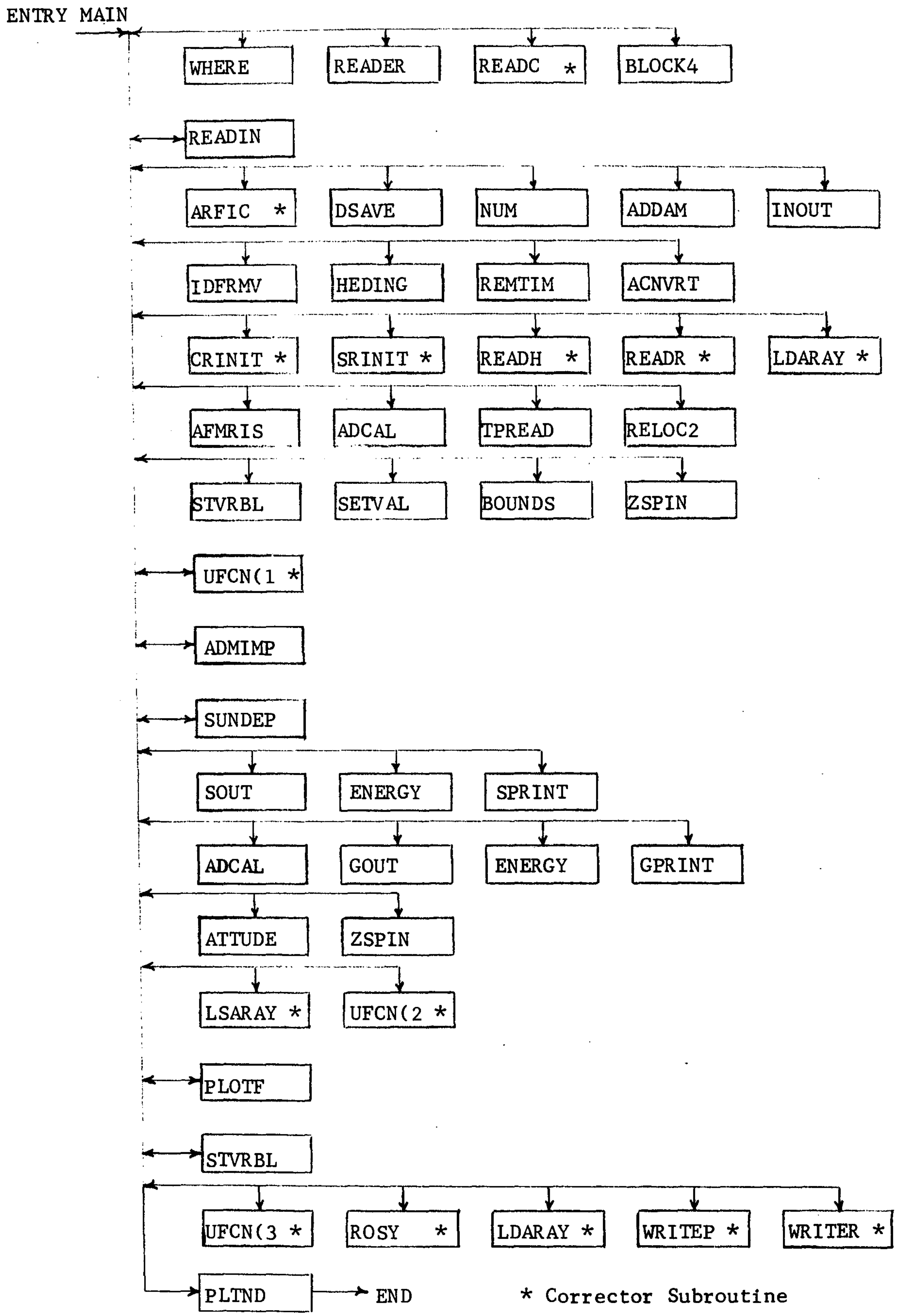


Figure 4 - MAIN Block Diagram With Corrector Added

SUBROUTINE NAME: ARFIC

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To preset control data when using standard flight data sets

CALLING SEQUENCE: Call ARFIC

LABELED COMMONS USED: ICNCØR, ICRPIN, IMAIN1, IRHIST, ISDARR, ISIMDT, IVARYP, XDBUG

SUBROUTINES CALLED: No subroutines

GLOSSARY OF VARIABLES: All variables defined in labeled common

SUBROUTINE NAME: BLØCK4

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To preset data

CALLING SEQUENCE: Call BLØCK4

LABELED COMMONS USED: PRCOM, IVARYP, EVARYP, BCNVRT, XDBUG, ISIMDT, IRHIST, IRØSIE, CØUFCN, CRØSIE, ICNCØR, CROSY, TWRIT

SUBROUTINES CALLED: No subroutines

GLOSSARY OF VARIABLES: All variables defined in labeled common

SUBROUTINE NAME: CRINIT

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: This subroutine sets up the vector of parameters P(I) to be varied and the vector of initial variations E(I)

CALLING SEQUENCE: Call CRINIT

LABELED COMMONS USED: IVARYP, EVARYP, XIN1, RPOOL1, XIN2, CSTVAL, ANTENA, XIN3, CCNVRT, RTDIST, CSOLAR, CFNALP, CDMPER, RDAMPR, XDBUG, CFNDGB, ICRPIN, PARRAY, BCNVRT, ICNCØR, IMAIN1, IPØØL1

SUBROUTINES CALLED:

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	DIMENSION	TYPE	DESCRIPTION
X01(3)			R*8	Substitution symbol for PSI1, THET1, PHI1
X02(3)			R*8	Substitution symbol for ØMEG(3)
X03(6)			R*8	Substitution symbol for ALFAE, BETAE, GAMAE, ØMBC(3)
X04(4)			R*8	Substitution symbol for BETLD, GAMLD, PHILD, DPHILD
X05(120)			R*8	Substitution symbol for A(10,3), ADØT(10,3), B(10,3), BDØT(10,3)
X06(120)			R*8	Substitution symbol for DIN(10,3), DINDØT(10,3), DØUT(10,3), DØUTDT(10,3)
X07(30)			R*8	Substitution symbol for ALFAEK(10), BETAEK(10), GAMAEK(10)

SUBROUTINE NAME: CRINIT (Concl'd)

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE:

CALLING SEQUENCE:

LABELED COMMONS USED:

SUBROUTINES CALLED:

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	DIMENSION	TYPE	DESCRIPTION
X08(10)			R*8	Substitution symbol for EMØDLS(10)
X09(10)			R*8	Substitution symbol for TDIS(10)
X10(10)			R*8	Substitution symbol for ZLO(10)
X11(60)			R*8	Substitution symbol for SKØA(10,3), SKØB(10,3)
X12(6)			R*8	Substitution symbol for DECAY, ZMDØ, ZMDBØ, ZK1D, ZK2D, PHIS
X13			R*8	Substitution symbol for CNV
X14			R*8	Substitution symbol for DTØØ
X15			R*8	Substitution symbol for CDAMP(3,10)
X16			R*8	Substitution symbol for BSTIF

SUBROUTINE NAME: GRAM (C,D,NMAX)

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To calculate the transformation (C) which relates the 'new' search axes to the original parameter axes

CALLING SEQUENCE: Call GRAM (C,D,NMAX)

LABELED COMMONS USED: No labeled common

SUBROUTINES CALLED: No subroutines called

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	DIMENSION	TYPE	DESCRIPTION
C	$\left[c_{1n}\right]$		R*8	Transformation matrix to transform parameter changes on 'new' search axes to the original parameter axes
D	$\left\{d_n\right\}$		R*8	Accumulated parameter changes on last completed stage
NMAX			I*4	The number of parameters in the search vector

SUBROUTINE NAME: LDARAY

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To load the new parameter value into appropriate common array at the completion of a trial and

CALLING SEQUENCE: Call LDARAY

LABELED COMMONS USED: XIN1, RPØØL1, XIN2, XDBUG, CSTVAL, ANTENA, XIN3, CCNVRT, RTDIST, CSØLAR, CFNALP, BCNVRT, CDMPER, RDAMPR, CFNDGB, ICRPIN, PARRAY, ICNCØR

SUBROUTINES CALLED: No subroutines

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	DIMENSION	TYPE	DESCRIPTION
X01(3)			R*8	Substitution symbol for PSI1, THET1, PHI1
X02(3)			R*8	Substitution symbol for ØMEG(3)
X03(6)			R*8	Substitution symbol for ALFAE, BETAE, GAMAE, ØMBC(3)
X04(4)			R*8	Substitution symbol for BETLD, GAMLD, PHILD, DPHILD
X05(120)			R*8	Substitution symbol for A(10,3), ADØT(10,3), B(10,3), BDØT(10,3)
X06(120)			R*8	Substitution symbol for DIN(10,3), DINDØT(10,3), DØUT(10,3), DØUTDT(10,3)
X07(30)			R*8	Substitution symbol for ALFAEK(10), BETAEK(10), GAMAEK(10)

SUBROUTINE NAME: LDARAY (Concl'd)

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE:

CALLING SEQUENCE:

LABELED COMMONS USED:

SUBROUTINES CALLED:

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	DIMENSION	TYPE	DESCRIPTION
X08(10)			R*8	Substitution symbol for EMØDLS(10)
X09(10)			R*8	Substitution symbol for TDIS(10)
X10(10)			R*8	Substitution symbol for ZLO(10)
X11(60)			R*8	Substitution symbol for SKØA(10,3), SKØB(10,3)
X12(6)			R*8	Substitution symbol for DECAY, ZMDØ, ZMDBØ, ZK1D, ZK2D, PHIS
X13			R*8	Substitution symbol for CNV
X14			R*8	Substitution symbol for DTØØ
X15(30)			R*8	Substitution symbol for CDAMP(3,10)
X16			R*8	Substitution symbol for BSTIF

SUBROUTINE NAME: LSARAY

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To extract simulator output from common and assign it to appropriate data channel

CALLING SEQUENCE: Call LSARAY

LABELED COMMONS USED: CANTNA, CØNSTS, ISDARR, RPØØL1, SIMDAT, XDBUG, XIN1, XIN2

SUBROUTINES CALLED: No subroutines called

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	DIMENSION	TYPE	DESCRIPTION
XO1(3)			R*8	Substitution symbol for PSI1, THET1, PHI1
XO2(3)			R*8	Substitution symbol for ØMEG(3), in degrees/sec
XR2(3)			R*8	Substitution symbol for ØMEG in rad/sec
XO3(6)			R*8	Substitution symbol for ALFAE, BETAE, GAMAE, ØMBC(3)
XO4(2)			R*8	Substitution symbol for PHI, PHID in deg, deg/sec
XR4(2)			R*8	Substitution symbol for PHI, PHID in rad, rad/sec
XO5(120)			R*8	Substitution symbol for A(10,3), ADØT(10,3), B(10,3), BDØT(10,3)
XO6(120)			R*8	Substitution symbol for DIN(10,3), DINDØT(10,3), DØUT(10,3), DØUTDT(10,3)

SUBROUTINE NAME: READC

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To prepare input words to be read by READIN using the subroutine SETUP

CALLING SEQUENCE: Call READC

LABELED COMMONS USED: IVARYP, EVARYP, BCNVRT, ISIMDT, IRHIST, XDBUG, IRØSIE, CØUFCN, CRØSIE, ICNCØR, TWRIT

SUBROUTINES CALLED: SETUP

GLOSSARY OF VARIABLES: All variables defined in labeled common

SUBROUTINE NAME: READH

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To read flight data sets and store the pertinent data in a data array

CALLING SEQUENCE: Call READH

LABELED COMMONS USED: CSDATA, IRHIST, XDBUG

SUBROUTINES CALLED: ACNVRT

GLOSSARY OF VARIABLES: All variables defined in labeled common

SUBROUTINE NAME: RØSY

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: This subroutine evaluates the success or failure of a trail and provides stopping logic

CALLING SEQUENCE: Call RØSY

LABELED COMMONS USED: CØRØSY, CØUFCN, CRØSIE, CRØSY, ICNCØR, IRØSIE, PARRAY, RØSY1

SUBROUTINES CALLED: GRAM

GLOSSARY OF VARIABLES: All variables defined in labeled common

SUBROUTINE NAME: SRINIT

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To set up an indexing array to extract the appropriate simulator output which is to be compared with real data to determine performance

CALLING SEQUENCE: Call SRINIT

LABELED COMMONS USED: ISDARR, ISIMDT, XDBUG, IRØSIE, IMAIN1, ANTENA

SUBROUTINES CALLED: No subroutines

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	DIMENSION	TYPE	DESCRIPTION
IS1(3)			I*4	Substitition symbol for ISATT(3)
IS2(3)			I*4	Substitution symbol for ISATTR(3)
IS3(2)			I*4	Substitution symbol for ISPHID, ISPHI
IS4(120)			I*4	Substitution symbol for ISA(10,3), ISADØT(10,3), ISB(10,3), ISBDØT(10,3)
IS5(120)			I*4	Substitution symbol for ISDIN(10,3), ISDIND(10,3), ISDØUT(10,30), ISDØUD(10,3)

SUBROUTINE NAME: UFCN (ICHECK, TIME)

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To compute the performance criteria for each data channel and the total performance criteria for the system

CALLING SEQUENCE: Call UFCN (ICHECK, TIME)

LABELED COMMONS USED: IRHIST, CSDATA, SIMDAT, XDBUG, IRØSIE, PARRAY, PRØSIE, CØUFCN, IØUFCN, RMAIN1

SUBROUTINES CALLED: No subroutines

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	DIMENSION	TYPE	DESCRIPTION
ICHECK			I*4	ICHECK 1 Initialize and preset values at start of trial ICHECK 2 Compute individual performance criteria for each data channel at each data point in a trial ICHECK 3 Compute total performance criteria at the end of a trial
TIME	t	sec	R*8	Time in seconds from the start of the year

SUBROUTINE NAME: WRITEP

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To write standard corrector output
at the end of each trial

CALLING SEQUENCE: Call WRITEP

LABELED COMMONS USED: CØUFCN, PRØSIE, CRØSY, IRØSIE, IRHIST,
ICNCØR

SUBROUTINES CALLED: No subroutines

GLOSSARY OF VARIABLES: All variables defined in labeled common

SUBROUTINE NAME: WRITER (ICS, IOS) ENTRY READR

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: WRITER (ICS, IØS) Compute remaining CPU and IØ time and write corrector restart tape. READR to read corrector restart tape

CALLING SEQUENCE: Call WRITER (ICS, IØS) or Call READR

LABELED COMMONS USED: CØRØSY, CSDATA, ROSY1, XDBUG, ISDARR, ICRPIN, PARRAY, CRØSY

SUBROUTINES CALLED: REMTIM MAXO EXIT

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	DIMENSION	TYPE	DESCRIPTION
ICS			I*4	CPU time left at the start of the trial
IØS			I*4	IØ time left at the start of the trial
ICST			I*4	CPU time left at the end of the trail
IØST			I*4	IØ time left at the end of the trial

3.2 Definition Of Labeled Common

The new labeled common is defined in the following paragraphs. The interaction between new labeled common and subroutines is shown in Figure 5. A description of the modifications to existing common is given below.

CØMMØN/PRCØM/

The array STØRE (10, 30) was reduced to this size to conserve space.

CØMMØN/CØMSØL/

The subscripted internal force coefficients and temperature coefficients were removed to conserve space.

BLANK COMMON

COMMON/CO(7000)

This common was labeled COMMON/CØRBIT/CO(7000) for ease of insertion in segment 4.

COMMON/IO(300)

This common was labeled COMMON/IØRBIT/IØ(300) for ease of insertion in segment 4.

SUBROUTINES

LABELED COMMON	ARFIC	BLOCK4	CRINIT	GRAM	LDARAY	LSARAY	READC	READH	ROSY	SRINIT	UFCN	WRITEP	WRITER	CONVRT	MAIN	GOUT	GPRINT	SOUT
BCNVRT		X			X		X							X				
COROSY									X				X					
COROUT															X			
COUFCN		X					X		X		X	X						
CROSIE		X					X		X									
CROSY		X							X			X	X			X	X	X
CSDATA								X			X							
EVARYP		X	X				X											
ICNCOR	X	X	X		X		X		X			X						
ICRPIN	X		X		X								X					
IOUFCN											X							
IRHIST	X	X									X	X						
IROSIE		X					X		X	X	X	X			X			
ISDARR	X					X				X			X					
ISIMDT	X	X					X			X								
IVARYP	X	X	X				X											
PARRAY									X		X				X			
PROSIE											X	X						
ROSY1									X				X					
SIMDAT						X					X							
TWRIT		X					X								X	X	X	X
XDBUG	X	X	X		X	X	X	X		X	X		X		X		X	

Figure 5 - Labeled Common - SUBROUTINE Interaction

NAME: COMMON/BCNVRT/BSTIF

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To transfer bending stiffness coefficient

USED IN SUBROUTINES: BLOCK4 , CONVRT, CRINIT, LDARAY, READC

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
BSTIF		1.0 DO	N.D.	R*8	A coefficient to provide for variation in the bending stiffness of all elements simultaneously

NAME: COMMON/CSDATA/A(5000)

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To transfer flight data

USED IN SUBROUTINES: READH, UFCN, WRITER

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
A(5000)	--	--	--	R*8	Storage area for flight data read from flight data tape

NAME: COMMON/CRØSIE/CALPHA, CBETTA, CGAMMA, RATU, DEL, TØL

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To transfer search constants

USED IN SUBROUTINES: BLØCK4 , RØSY, READC

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
CALPHA	α	2.0 DO		R*8	Factor for parameter step in parameter search
CBETTA	β	0.5 DO		R*8	Factor for parameter step in parameter search
CGAMMA	γ	0.5 DO		R*8	Factor for parameter step in parameter search
RATU	R	0.5 DO		R*8	Stopping criteria
DET	D	0.2 DO		R*8	Stopping criteria
TOL		1.01 DO		R*8	Tolerance for trial success

NAME: COMMON/CRØSY/SUCESS, NTRIA, NSTAG, NSUCC

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To transfer control information for optimizing performance

USED IN SUBROUTINES: BLØCK4 , GØUT, GPRINT, SØUT, WRITEP, WRITER

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
SUCESS			N.D.	R*8	Control constant for indicating successful completion of search
NTRIA		0	N.D.	I*4	The number of the current trial
NSTAG			N.D.	I*4	The number of the current stage
NSUCC			N.D.	I*4	The number of consecutive successes

NAME: COMMON/CØUFCN/TWRDD, CWEGHT(20), CNORM(20), U, RESDUL(20)

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To transfer data relative to the computation of the performance criteria and the result

USED IN SUBROUTINES: BLØCK4 , READC, UFCN, WRITEP, RØSY

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
TWRDD	N.A.	10.0 D-38	N.D.	R*8	Test word for recognizing missing data
CWEGHT(20)		1.0 DO	N.D.	R*8	Weighting factor for residuals for each data channel
CNORM(20)		1.0 DO		R*8	Normalization constant for residuals for each data channel
U		--	--	R*8	The performance criteria computed for each trial
RESDUL(20)					The residual for each data channel

NAME: COMMON/CØRØUT/PHSAVE(12)

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: Not necessary

USED IN SUBROUTINES: MAIN

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
PHSAVE(12)		N.A.	N.A.	R*8	The current value of the parameters

NAME: COMMON/CØRØSY/C(12,12), D(12), DP(12), ULAST, USTAG, UPREV

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To transfer current basic corrector data

USED IN SUBROUTINES: RØSY, WRITER

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
C(12,12)		N.A.	N.A.	R*8	Transformation from search directions to parameter space
D(12)		N.A.	N.A.	R*8	Total change in search direction for this stage
DP(12)		N.A.	N.A.	R*8	Increment in parameter for next trial
ULAST		N.A.	N.A.	R*8	Performance criterion at last successful trial
USTAG		N.A.	N.A.	R*8	Performance criterion at end of last stage
UPREV		N.A.	N.A.	R*8	Performance criterion at end of second last stage

NAME: COMMON/EVARYP/EVATT, EVATTR, EVPHIL, EVPHID, EVABD(3), EVABDD(3), EVALPK, EVBETK, EVGAMK, EVEMØD, EVTDIS, EVZLO, EVSKØ(3), EVDCAY, EVZMDØ, EVZMDB, EVZKID, EVZK2D, EVPHIS, EVCNV, EVDTØØ, EVCDMP(3), EVSTIF

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE; To transfer initial parameter variations

USED IN SUBROUTINES: BLØCK4 , CRINIT, READC

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
EVATT		0.0 DO	deg	R*8	Initial variation for attitude angle initial conditions
EVATTR		0.0 DO	deg/sec	R*8	Initial variation for attitude angular rate initial conditions
EVPHIL		0.0 DO	deg	R*8	Initial variation for damper reference frame angles BETLD, GAMLD, PHILD
EVPHID		0.0 DO	deg	R*8	Initial variation for damper angular rate initial condition DPHILD
EVABD(3)		0.0 DO	feet	R*8	Initial variation for element displacement initial conditions A(10,3), B(10,3), DIN(10,3), DØUT(10,3), subscripted to mode

NAME: COMMON/EVARYP/(Cont'd)

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE:

USED IN SUBROUTINES:

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
EVABDD(3)		0.0 DO	feet/sec	R*8	Initial variation for element velocities ADØT(10,3), BDØT(10,3), DINDØT(10,3), DØUTDT(10,3). Subscripted to mode
EVALPK		0.0 DO	deg	R*8	Initial variation for element frame Euler angle ALFAEK
EVBETK		0.0 DO	deg	R*8	Initial variation for element frame Euler angle BETAEK
EVGAMK		0.0 DO	deg	R*8	Initial variation for element frame Euler angle GAMAEK
EVEMØD		0.0 DO	lbs/in^2	R*8	Initial variation for element Young's modulus EMØDLS
EVTDIS		0.0 DO	N.D.	R*8	Initial variation for element temperature differential TDIS

NAME: COMMON/EVARYP/(Cont'd)

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE:

USED IN SUBROUTINES:

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
EVZLO		0.0 DO	feet	R*8	Initial variation for element length ZLO
EVDCAY		0.0 DO	N.D.	R*8	Initial variation for damper suspension magnetic hysteresis decay factor DECAY
EVZMDØ		0.0 DO	foot lbs	R*8	Initial variation for damper suspension magnetic hysteresis saturation moment ZMDØ
EVZMDB		0.0 DO	foot lbs	R*8	Initial variation for damper suspension magnetic hysteresis initial magnetic moment ZMDBØ
EVZK1D		0.0 DO	foot lbs	R*8	Initial variation for damper suspension spring constant ZK1D
EVZK2D		0.0 DO	foot lbs	R*8	Initial variation for damper suspension stop spring constant ZK2D

NAME: COMMON/EVARYP/(Concl'd)

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE:

USED IN SUBROUTINES:

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
EVPHIS		0.0 DO	deg	R*8	Initial variation for damper suspension stop angle PHIS
EVCNV		0.0 DO	foot lb sec	R*8	Initial variation for damper suspension viscous damping coefficient CNV
EVDTØØ		0.0 DO	°F	R*8	Initial variation for element temperature differential DTØØ
EVCDMP(3)		0.0 DO		R*8	Initial variation for element damping factor CDAMP(3,10), subscripted to mode
EVSTIF		0.0 DO	N.D.	R*8	Initial variation for overall bending stiffness coefficient BSTIF

NAME: COMMON/ISIMDT/ISATT(3), ISATTR(3), ISPHID, ISPHI, ISA(10,3), ISADØT(10,3), ISB(10,3), ISBDØT(10,3), ISDIN(10,3), ISDIND(10,3), ISDØUT(10,3), ISDØUD(10,3)

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To transfer control data for constructing simulator data array

USED IN SUBROUTINES: ARFIC, BLOCK4 , READC, SRINIT

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
ISATT(3)		0	N.D.	I*4	Control word for assigning simulator attitude data to data channels
ISATTR(3)		0	N.D.	I*4	Control word for assigning simulator attitude rate data to data channels
ISPHID		0	N.D.	I*4	Control word for assigning simulator damper angular velocity to data channel
ISPHI		0	N.D.	I*4	Control word for assigning simulator damper angular position to data channel
ISA(10,3)		0	N.D.	I*4	Control word for assigning simulator element A(10,3) displacement to data channel
ISADØT(10,3)		0	N.D.	I*4	Control word for assigning simulator element ADØT(10,3) velocity to data channel

NAME: COMMON/ICRPIN/IPAR(12,2)

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To transfer control data for transferring new parameters into common

USED IN SUBROUTINES: ARFIC, CRINIT, LDARAY, WRITER

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
IPAR(12,2)				I*4	This array of integers defines the common locations of the parameters being varied in the optimization process

NAME: COMMON/ISDARR/ISAR(20,2)

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To transfer control data for loading simulator output into data channels

USED IN SUBROUTINES: ARFIC, LSARAY, SRINIT, WRITER

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
ISAR(20,2)				I*4	This array of integers provides information necessary to extract simulator output from common and load it into the appropriate channel of the simulator data vector

NAME: COMMON/ICNCØR/ICØRIC, NPARAM

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE:

USED IN SUBROUTINES: ARFIC, BLØCK4 , CRINIT, LDARAY, READC, ROSY, WRITEP

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
ICORIC		1		I*4	Control word for bypassing initial condition variations NOTE: Do not use
NPARAM		12		I*4	The number of parameters to be varied in a corrector run $1 \leqslant NPARAM \leqslant 12$

NAME: COMMON/IRH1ST/NDWORD, ITWRD(2), IUWORD(20), IDCHAN

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To transfer control data for reading flight data sets

USED IN SUBROUTINES: ARFIC, BLØCK4 , READC, READH, UFCN, WRITEP

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
NDWORD		300		I*4	The number of real*8 words in a flight data set data record
ITWORD(2)		(1) 1 (2) 2		I*4	The location in the data record of the two time words date, and time of day
IUWORD(20)		(1) 21 (2) 22 (3) 23 (4-20) 0		I*4	The location in the data record of the data to be assigned to each data channel
IDCHAN		3		I*4	The number of data channels to be used

NAME: COMMON/IVARYP/IVATT(3), IVATTR(3), IVBELD, IVGALD, IVPHI, IVPHID, IVA(10,3), IVADØT(10,3), IVB(10,3), IVBDØT(10,3), IVDIN(10,3), IVDIND(10,3), IVDØUT(10,3), IVDØUD(10,3), IVALPK(10), IVBETK(10), IVGAMK(10), IVEMØD(10), IVTDIS(10), IVZLO(10), IVSKØA(10,3), IVSKØB(10,3), IVDCAY, IVZMDØ, IVZMDB, IVZK1D, IVZK2D, IVPHIS, IVCNV, IVDTØØ, IVCDMP(3,10), IVSTIF

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To transfer control data defining parameters to be varied

USED IN SUBROUTINES: ARFIC, BLOCK , CRINIT, READC

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
IVATT(3)		0		I*4	Control word to vary attitude angle initial conditions
IVATTR(3)		0		I*4	Control word to vary attitude rate initial conditions
IVBELD		0		I*4	Control word to vary damper frame angle BETLD
IVGALD		0		I*4	Control word to vary damper frame angle GAMLD
IVPHI		0		I*4	Control word to vary damper motion initial angle PHILD
IVPHID		0		I*4	Control word to vary damper motion initial angular velocity DPHILD
IVA(10,3)		0		I*4	Control word to vary element displacement initial conditions A(10,3)

NAME: COMMON/IVARYP/(Cont'd)

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE:

USED IN SUBROUTINES:

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
IVADØT(10,3)		0		I*4	Control word to vary element velocity initial conditions ADØT(10,3)
IVB(10,3)		0		I*4	Control word to vary element displacement initial conditions B(10,3)
IVBDØT(10,3)		0		I*4	Control word to vary element velocity initial conditions BDØT(10,3)
IVDIN(10,3)		0		I*4	Control word to vary damper element initial displacement DIN(10,3)
IVDIND(10,3)		0		I*4	Control word to vary damper element initial velocity DINDØT(10,3)
IVDØUT(10,3)		0		I*4	Control word to vary damper element initial displacement DØUT(10,3)

NAME: COMMON/IVARYP/(Cont'd)

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE:

USED IN SUBROUTINES:

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
IVDØUD(10,3)		0		I*4	Control word to vary damper element initial velocity DØUTDT(10,3)
IVALPK(10)		0		I*4	Control word to vary element frame Euler angle ALFAEK(10)
IVBETK(10)		0		I*4	Control word to vary element frame Euler angle BETAEK(10)
IVGAMK(10)		0		I*4	Control word to vary element frame Euler angle GAMAEK(10)
IVEMØD(10)		0		I*4	Control word to vary element Young's modulus EMØDLS(10)
IVTDIS(10)		0		I*4	Control word to vary temperature differential TDIS(10)
IVZLO(10)		0		I*4	Control word to vary element length ZLO(10)

NAME: COMMON/IVARYP/(Cont'd)

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE:

USED IN SUBROUTINES:

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
IVSKØA(10,3)		0		I*4	Control word to vary element no load offset SKØA(10,3)
IVSKØB(10,3)		0		I*4	Control word to vary element no load offset SKØB(10,3)
IVDCAY		0		I*4	Control word to vary damper suspension magnetic hysteresis decay factor, DECAY
IVZMDØ		0		I*4	Control word to vary damper suspension magnetic hysteresis saturation moment, ZMDØ
IVZMDB		0		I*4	Control word to vary damper suspension magnetic hysteresis initial magnetic moment, ZMDBØ
IVZK1D		0		I*4	Control word to vary damper suspension spring constant, ZK1D

NAME: COMMON/IVARYP/(Concl'd)

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE:

USED IN SUBROUTINES:

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
IVZK2D		0		I*4	Control word to vary damper suspension stop spring constant, ZK2D
IVPHIS		0		I*4	Control word to vary damper suspension stop angle, PHIS
IVCNV		0		I*4	Control word to vary damper suspension viscous damping coefficient, CNV
IVDTØØ		0		I*4	Control word to vary solar temperature gradient on elements, DTØØ
IVCDMP(3,10)		0		I*4	Control word to vary element damping coefficient, CDAMP(3,10)
IVSTIF		0		I*4	Control word to vary bending stiffness coefficient for all elements, BSTIF

NAME: COMMON/IRØSIE/MAXTRY, NWT, MMAG

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To transfer control words for performance criteria

USED IN SUBROUTINES:

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
MAXTRY		100		I*4	Maximum number of trials
NWT				I*4	Control word to specify weighting and normalization of residuals
MMAG				I*4	Control word to specify type of residuals, i.e., absolute value or squares

NAME: COMMON/IØUFCN/IPU(20), IEUFCN

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE:

USED IN SUBROUTINES: **UFCN**

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
IPU(20)		--		I*4	Counter for bad data
IEUFCN				I*4	Spare

NAME: COMMON/PARRAY/P(12), E(12)

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To transfer parameter vector and initial variation vector

USED IN SUBROUTINES: CRINIT, MAIN, LDARAY, RØSY, UFCN, WRITER

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
P(12)				R*8	Vector of parameter values in the variation process
E(12)				R*8	Vector of initial parameter variations

NAME: COMMON/PRØSIE/P(12)

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To transfer current array of parameters for output

USED IN SUBROUTINES: UFCN, WRITEP

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
P(12)				R*8	Vector of parameters used in the current trial of the optimization

NAME: COMMON/RØSY1/N

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To transfer data for restart tape

USED IN SUBROUTINES: ROSY, WRITER

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
N				I*4	The number of the current search axis $1 \leqslant N \leqslant NPARAM$

NAME: COMMON/SIMDAT/B(20)

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To transfer simulator data for calculating residuals

USED IN SUBROUTINES: LSARAY, UFCN

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
B(20)				R*8	Vector of current simulator data to be used in computing residuals at current data point

NAME: COMMON/TWRIT/NPRINT, NHISTS, JØPTIM, IREIN

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To transfer basic control words for corrector operation

USED IN SUBROUTINES: BLØCK4 , MAIN, GØUT, GPRINT, READC, SØUT

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
NPRINT		1	N.D.	I*4	Control word to suppress regular simulator output during corrector runs i.e. NPRINT 1 every trial printed NPRINT 10 first, eleventh, etc., trials printed only
NHISTS		0	N.D.	I*4	Control word to write out regular simulator run as a standard flight data set
JØPTIM		0	N.D.	I*4	Control word for corrector runs JØPTIM 0 regular simulator run JØPTIM 1 corrector run
IREIN		0	N.D.	I*4	Control word to read in restart tape IREIN 0 regular start IREIN 1 restart

NAME: COMMON/XDBUG/IDBUG, IFLITE, IZEF3, IZEF4

LANGUAGE: FORTRAN IV

MACHINE: IBM 360/75,91,95 (OS)

PURPOSE: To transfer special control words

USED IN SUBROUTINES: ARFIC, BLOCK4 , CRINIT, MAIN, GPRINT, LDARAY, LSARAY, READC, READH, SRINIT, UFCN, WRITER

GLOSSARY OF VARIABLES:

FORTRAN SYMBOL	MATH SYMBOL	PRESET VALUE	UNITS	TYPE	DESCRIPTION
IDBUG		0	N.D.	I*4	Control word to write out special debugging output for corrector
IFLITE		0	N.D.	I*4	Control word to write out general flight data set
IZEF3		0	N.D.	I*4	Spare
IZEF4		0	N.D.	I*4	Spare

3.3 Definition of Input

The new and modified input symbols are defined in the following paragraphs.

3.3.1 Generation of Test Flight Data Sets

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	UNITS
NHISTS		Control word for generating Standard Test Flight Data Set. NHISTS 0 No data set generated NHISTS 1 Standard data set generated depending on INØPT NHISTS 1 INØPT 1 each data Record contains 5 double precision words Date time PSI1 PHI1 THET1 NHISTS 1 INØPT 2 each data Record contains 14 double precision words DATE TIME ALFAE BETAE GAMAE A(1,1) A(2,1) A(3,1) A(4,1) B(1,1) B(2,1) B(3,1) B(4,1) PHILD	Integer N.D.
IFLITE		Control word for generating General Test Flight Data Set IFLITE 0 No data set generated IFLITE 1 General data set generated by writing out complete normal output data record from GPRINT. Each data record contains store(300) Note: NHISTS 0 When IFLITE 1 and IFLITE 0 When NHISTS 1	Integer N.D.

3.3.2 General Control Words

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	UNITS
JØPTIM		Control word for corrector run JØPTIM 0 Normal simulator run JØPTIM 1 Corrector run	Integer N.D.
NPARAM		Control word specifying the number of parameters to be varied in optimising the fit to flight data NPARAM 1 One Parameter NPARAM 2 Two Parameters Etc. Note: $1 \leqslant \text{NPARAM} \leqslant 12$	Integer N.D.
J		Control word specifying the type of performance criterion to be used	Integer N.D.

$$J = 1 \qquad U = \sum_i \sum_j |\delta_{ij}|$$

$$J = 2 \qquad U = \sum_i \sum_j (\delta_{ij})^2$$

$$J = 3 \qquad U = \sum_i \sum_j |\delta'_{ij}|$$

$$J = 4 \qquad U = \sum_i \sum_j (\delta_{ij})^2$$

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	UNITS
MAXTRY		The maximum number of trials to be performed in optimising the fit to the flight data	Integer N.D.
NPRIN		Control word to suppress standard simulator output for every trial For instance; NPRIN 10 gives regular simulator output at the first trial and every 10th trial thereafter NPRIN 1 gives regular simulator output for every trial	Integer N.D.
IDBUG		Control word for corrector DEBUG output. <u>Not normally used</u> IDBUG 0 No Output IDBUG 1 Debug Output	Integer N.D.

3.3.3 Specification of parameters to be varied

The user specifies the order in which parameters are to be varied by assigning the appropriate integer to the parameter control word; 1 is varied first; 2 is varied second; etc. Note: no more than 12 parameters can be varied in a single corrector run.

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	UNITS
IVATT(3)		Control word for varying initial Conditions for attitude angles INØPT 1 IVATT(1) - PSI1 IVATT(2) -THET1 IVATT(3) - PHI1 INØPT 2 IVATT(1) - ALFAE IVATT(2) - BETAE IVATT(3) - GAMAE	Integer N.D.
IVATTR(3)		Control word for varying initial conditions for attitude angular rates INØPT 1 IVATTR (1) - ØMEG(1) IVATTR (2) - ØMEG(2) IVATTR (3) - ØMEG(3) INØPT 2 IVATTR (1) - ØMBC(1) IVATTR (2) - ØMBC(2) IVATTR (3) - ØMBC(3)	Integer N.D.
IVBELD		Control word for varying damper frame angle BETLD	Integer N.D.

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	UNITS
IVGALD		Control word for varying the damper frame angle GAMLD	Integer N.D.
IVPHI		Control word for varying the initial condition on the damper angular displacement PHILD	Integer N.D.
IVPHID		Control word for varying the initial condition on the damper angular velocity DPHILD	Integer N.D.
IVA(10,3)		Control word for varying the initial conditions on the element A displacements	Integer N.D.
IVADØT(10,3)		Control word for varying the initial conditions on the element ADOT velocities	Integer N.D.
IVB(10,3)		Control word for varying the initial conditions on the element B displacements	Integer N.D.
IVDIN(10,3)		Control word for varying the initial conditions on the damper element DIN displacements	Integer N.D.
IVDIND(10,3)		Control word for varying the initial conditions on the damper element DINDØT velocities	Integer N.D.
IV DØUT(10,3)		Control word for varying the initial conditions on the damper element DØUT displacements	Integer N.D.
IVDØUD(10,3)		Control word for varying the initial conditions on the damper element DØUTDT velocities	Integer N.D.
IVALPK(10)		Control word for varying the element frame EULER angle ALFAEK	Integer N.D.

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	UNITS
IVBETK(10)		Control word for varying the element frame euler angle BETAEK	Integer N.D.
IVGAMK(10)		Control word for varying the element frame euler angle GAMAEK	Integer N.D.
IVBDØT(10,3)		Control word for varying the initial conditions on the element BDOT velocities	Integer N.D.
IVEMØD(10)		Control word for varying Young's modulus, EMØDLS, for the element	Integer N.D.
IVTDIS(10)		Control word for varying the temperature differential across the element TDIS	Integer N.D.
IVZLO(10)		Control Word for varying the element length ZLO	Integer N.D.
IVSKØA(10,3)		Control word for varying the no load offset for the element in the A direction SKØA	Integer N.D.
IVSKØB(10,3)		Control word for varying the no load offset for the element in the B direction SKØB	Integer N.D.
IVDCAY		Control word for varying the exponential decay constant in the magnetic hysteresis damper simulate. DECAY	Integer N.D.
IVZMDØ		Control word for varying the saturation torque in the magnetic hysteresis damper simulation ZMDØ	Integer N.D.
IVZMDB		Control word for varying the initial magnetic torque in the magnetic hysteresis damper simulation. ZMDBØ	Integer N.D.
IVZK1D		Control word for varying the damper suspension spring constant ZK1D	Integer N.D.

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	UNITS
IVZK2D		Control word for varying the damper suspension stop spring constant ZK2D	Integer N.D.
IVPHIS		Control word for varying the damper suspension stop angle PHIS	Integer N.D.
IVCNV		Control word for varying the damper suspension viscous torque coefficient CNV	Integer N.D.
IVDTØØ		Control word for varying the temperature differential across all elements simultaneously DTØØ	Integer N.D.
IVSTIF		Control word for varying the bending stiffness of all elements simultaneously BSTIF	Integer N.D.
IVCDMP(3,10)		Control word for varying the damping factor for antenna motions CDAMP	Integer N.D.

3.3.4 Initial Variations for Parameters

The user must specify an initial variation step for the parameters to be varied in a corrector run.

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	UNITS
EVATT		Initial step size for variations in attitude angle initial conditions. Must be specified if IVATT(1-3) non zero	Degrees
EVATTR		Initial step size for variations in attitude angular rate initial conditions. Must be specified if IVATTR (1-3) non zero	Degrees/sec
EVPHIL		Initial step size for variations in damper reference frame Euler angles BETLD GAMLD PHILD. Must be specified if IVBELD, IVGALD, or IVPHI non zero	Degrees

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	UNITS
EVPHID		Initial step size for variation in damper motion angular rate initial condition DPHILD. Must be specified if IVPHID non zero	Degrees/sec
EVABD(3)		Initial step size for variations in element displacement initial conditions. Must be specified if IVA(1-10,3), IVB(1-10,3), IVDIN(1-10,3), or IVDØUT(1-10,3) are non zero EVABD(1) for A(1-10,1), B(1-10,1), DIN(1-10,1), DØUT(1-10,1) EVABD(2) for A(1-10,2), B(1-10,2), DIN(1-10,2), DØUT(1-10,2) EVABD(3) for A(1-10,3), B(1-10,3), DIN(1-10,3), DØUT(1-10,3)	Feet
EVABDD(3)		Initial step size for variations in element velocity initial conditions. Must be specified if IVADØT(1-10,3), IVBDØT(1-10,3), IVDIND(1-10,3), or IVDØUD(1-10,3) are non zero EVABDD(1) for ADØT(1-10,1), BDØT(1-10,1) DINDØT(1-10,1), DØUTDT(1-10,1) EVABDD(2) for ADØT(1-10,2), BDØT(1-10,2) DINDØT(1-10,2), DØUTDT(1-10,2) EVABDD(3) for ADØT(1-10,3), BDØT(1-10,3) DINDØT(1-10,3), DØUTDT(1-10,3)	Feet/sec
EVALPK		Initial step size for variation in element reference frame Euler angle ALFAEK. Must be specified if IVALPK (1-10) non zero	Degrees
EVBETK		Initial step size for variation in element reference frame Euler angle BETAEK. Must be specified if IVBETK(1-10) non zero	Degrees
EVGAMK		Initial step size for variation in element reference frame Euler angle GAMAEK. Must be specified if IVGAMK(1-10) non zero	Degrees

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	UNITS
EVEMØD		Initial step size for variation in Young's modulus of element EMØDLS. Must be specified if IVEMØD(1-10) non zero	Pounds/inch^2
EVTDIS		Initial step size for variation in the temperature differential across the element TDIS. Must be specified if IVTDIS(1-10) non zero	N.D.
EVZLO		Initial step size for variation of element length ZLO. Must be specified if IVZLO(1-10) non zero	Feet
EVSKØ(3)		Initial step size for variation of element no load offsets SKØA, SKØB. Must be specified if IVSKØA(1-10,3) or IVSKØB(1-10,3) are non zero EVSKØ(1) for SKØA(1-10,1), SKØB(1-10,1) EVSKØ(2) for SKØA(1-10,2), SKØB(1-10,2) EVSKØ(3) for SKØA(1-10,3), SKØB(1-10,3)	Feet
EVDCAY		Initial step size for variation of the exponential decay constant in the magnetic hysteresis damper simulation. Must be specified if IVDCAY non zero	N.D.
EVZMDØ		Initial step size for variation of the saturation torque in the magnetic hysteresis damper simulation. Must be specified if IVZMDØ non zero	Foot pounds
EVZMDB		Initial step size for variation of the initial magnetic torque in the magnetic hysteresis damper simulation. Must be specified if IVZMDB non zero	Foot pounds
EVZK1D		Initial step size for variation of the damper suspension spring constant ZK1D. Must be specified if IVZK1D non zero	Foot pounds/rad

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	UNITS
EVZK2D		Initial step size for variation of the damper suspension stop spring constant ZK2D. Must be specified if IVZK2D non zero	Foot-pounds/ rad
EVPHIS		Initial step size for variation of the damper suspension stop angle PHIS. Must be specified if IVPHIS non zero	Degrees
EVCNV		Initial step size for variation of the damper suspension viscous torque coefficient CNV. Must be specified if IVCNV non zero	Foot-pounds/ sec
EVDTØØ		Initial step size for variation of the temperature differential across all elements simultaneously. Must be specified if IVDTØØ non zero	°F
EVSTIF		Initial step size for variation of the bending stiffness for all elements simultaneously. Must be specified if IVSTIF non zero	N.D.
EVCDMP(3)		Initial step size for variation of element damping coefficient EVCDMP(1) for CDAMP(1,1-10) EVCDMP(2) for CDAMP(2,1-10) EVCDMP(3) for CDAMP(3,1-10) Must be specified if IVCDMP(3,1-10) non zero	N.D.

3.3.5 Reading of Flight Data Sets

The input described below is required in the reading of general flight data sets only. Standard flight data sets, defined in the format by NHISTS and INØPT, will be read automatically by the program without special input.

The general flight data set is assumed to be a series of sequential data records with no header record. Each data record is assumed to be a vector of Real*8 words. Each data record is

assumed to contain the date (YYMMDD.) year, month, day, and the time of day (HHMMSS.) hour, minute, second.

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	UNITS
ISTACK		Control word for presetting data for the reading of standard flight data sets ISTACK 0 standard flight data ISTACK non zero general flight data set	Integer N.D.

The following input words are required only for ISTACK non zero:

NDWORD		The number of Real*8 words in each data record For general flight date sets generated for testing purposes with the program NDWORD 300	Integer N.D.
ITWRD(2)		The location in the data record of the two time words DATE = YYMMDD. HMS = HHMMSS. ITWRD(1) is location of DATE ITWRD(2) is location of HMS For general flight data sets generated for testing purposes with the program ITWRD(1) 1 ITWRD(2) 2	Integer N.D.
IDCHAN		The number of data channels to be used in the evaluation of the performance criteria. The maximum number of data channels is 20	Integer N.D.
IUWORD(20)		The location in the data record of the various data channels for instance, with a general flight data set generated for testing purposes with the program and INØPT 2 IUWØRD(1) 21 ALFAE in data channel 1 IUWØRD(2) 22 BETAE in data channel 2 IUWØRD(3) 23 GAMAE in data channel 3	Integer N.D.

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	UNITS
TWRDD		Test word for recognizing missing data	N.D.

3.3.6 Simulator Data Channels for Performance Criteria

The input required to assign the appropriate data channel to simulator output in order to construct the performance criteria is described below. In the case where the flight data set is of a standard format consistent with INØPT the simulator output is assigned automatically by use of ISTACK 0. For standard flight data sets the data channel assignments are as follows:

CHANNEL NO.	INØPT 1	INOPT 2
1	PSI1	ALFAE
2	PHI1	BETAE
3	THET1	GAMAE
4	UNUSED	A(1,1)
5	"	A(2,1)
6	"	A(3,1)
7	"	A(4,1)
8	"	B(1,1)
9	"	B(2,1)
10	"	B(3,1)
11	"	B(4,1)
12	"	PHILD

No additional input is required to assign the above formats using standard flight data sets other than ISTACK 0 and INØPT 1 or 2.

In the use of general flight data sets it is necessary to provide input to assign the appropriate simulator output quantities to the proper data channel. For general flight data sets the control word ISTACK must be non zero. The assignment of the flight data to particular data channels is made by the input of the vector IUWORD(20) in extracting the data from the general

flight data set. The assignment of the simulator output data to construct the performance criteria must be compatible with the data channels assigned for the flight data.

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	UNITS
ISATT(3)		Control word to assign simulator output to appropriate data channel. The meaning of ISATT(3) is modified by INØPT INØPT 1 ISATT(1) J assigns PSI1 to channel J ISATT(2) J assigns THET1 to channel J ISATT(3) J assigns PHI1 to channel J INØPT 2 ISATT(1) J assigns ALFAE to channel J ISATT(2) J assigns BETAE to channel J ISATT(3) J assigns GAMAE to channel J	Integer N.D.

NOTE: J = 0 data not assigned

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	UNITS
ISATTR(3)		Control word to assign simulator output to appropriate data channel. The meaning of ISATTR(3) is modified by INØPT INØPT 1 ISATTR(I) J assigns ØMEG(I) to channel J INØPT 2 ISATTR(I) J assigns ØMBC(I) to channel J	Integer N.D.
ISPHI		Control word to assign simulator output damper angle PHILD to appropriate data channel	Integer N.D.
ISPHID		Control word to assign simulator output damper angular rate DPHILD to appropriate data channel	Integer N.D.
ISA(10,3)		Control word to assign simulator output element displacement A(10,3) to appropriate data channel	Integer N.D.
ISADØT(10,3)		Control word to assign simulator output element velocity ADØT(10,3) to appropriate data channel	Integer N.D.
ISB(10,3)		Control word to assign simulator output element displacement B(10,3) to appropriate data channel	Integer N.D.

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	UNITS
ISBDØT(10,3)		Control word to assign simulator output element velocity BDØT(10,3) to appropriate data channel	Integer N.D.
ISDIN(10,3)		Control word to assign simulator output damper element displacement DIN(10,3) to appropriate data channel	Integer N.D.
ISDIND(10,3)		Control word to assign simulator output damper element velocity DINDØT(10,3) to appropriate data channel	Integer N.D.
ISDØUT(10,3)		Control word to assign simulator output damper element displacement DØUT(10,3) to appropriate data channel	Integer N.D.
ISDØUD(10,3)		Control word to assign simulator output damper element velocity DØUTDT(10,3) to appropriate data channel	Integer

3.3.7 Performance Criteria

The performance criteria is constructed using the residuals from each data channel either directly or weighted and normalized. The input symbols used to provide weighting and normalization are described below:

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	UNITS
CWEGHT(20)		Weighting factor for residuals subscripted to the data channel. These are used only with J = 3 or 4	N.D.
CNORM(20)		Normalization constant for residuals subscripted to the data channel. These are used only with J = 3 or 4	Units of Data

3.3.8 Input for Parameter Search Operation

In the process of minimizing the performance criteria certain constants are required in computing parameter variations and in testing the performance criteria. The input symbols for these constants are described below. However, the use of preset values is recommended for routine operation.

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	UNITS
CALPHA	α	Factor in selecting next step size of parameter after a successful trial $\alpha > 1$	N.D.
CBETTA	β	Factor in selecting next step size of parameter after an unsuccessful trial $\beta < 1$	N.D.
CGAMMA	γ	Factor in selecting next step size of parameter after a successful stage	N.D.
RATU	R	Stopping criteria constant ratio of difference in three successive stage performance criteria $0.0 < R < 1.0$	N.D.
DEL	D	Band in first completion criteria $0 < D < 1.0$	N.D.
TØL		Tolerance for determining success of a given trial	N.D.

3.3.9 Restart Procedure

Normal corrector operation to determine best fit parameters will require relatively large amounts of machine time. The program is designed to check remaining CPU and IO time at the end of each trial. When either remaining time approaches the maximum time for any trial a restart tape is written so that the search can be continued from that point automatically.

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	UNITS
		Control word to read a restart tape to continue a corrector run. IREIN 0 not a restart IREIN 1 read restart tape NOTE: To use a restart tape the original data deck must be used with the addition of IREIN 1	Integer N.D.

3.4 Description of Output

The normal corrector output is printed at the end of each trial. The first line consists of three control words which describe the state of the search procedure.

NSTAGE — The number of completed searches through the parameter space.

NTRIAL — The number of the trial. This counter is incremented after each trial.

NSUCCESS — The number of consecutive trials in any parameter direction in which the performance criteria was reduced. If NSUCCESS is non zero the current trial reduced the performance criteria.

The vector of parameter values for the current trial is printed out under the heading parameter values. The total performance criteria for the current trial is printed and the individual residuals of each data channel are recorded.

3.5 Overlay Structure

This section describes the organization of the program and the overlay structure. The program consists of four modules. A description of each module by size and composition is presented and the overlay tree structure is shown in Figure 6. A job control language listing is also included.

1. Root Segment A, Size 197 K

HAMØUT	DATØUT	SATPØS	ITEMP2	EISUBK	ØUTØNE	RMAIN2
IDATE1	CCMBNZ	ZSPINR	XIN4	XIN3	JBTEST	ITEMP
RPØØL5	HEAD22	RPØØL3	RPØØL1	RFASTR	RDAMPR	IØDPLY
XIN2	XIN1	SWITCH	RTDIST	RSUNCL	IFASTR	IMAIN2
GLITCH	DEPLØY	CSTVAL	CØDPLY	CFINDX	CDMPER	ADREAL
PLØTIT	NØDER	RVISCS	RMGNTC	ADDAM	ACNVRT	ARFDT2
ARTLU1	CHECK	CØMP62	DRØUT	ELØUT	H1	HMSØUT
INØUT	MULTM	QTIME	STDPND	STVRBL	TCNVRT	YRMØDA
NUM	FVAL	LA000000	MAIN	SBLOGOR	ANUMARG	DECK1
TYPSET	SHFT2V	ADCN	BCDWD	CØRE	CSDATA	PRØSIE
SUNANG	IØUFCN	ICRPIN	PARRAY	ICNCØR	IRØSIE	ISIMDT
EVARYP	IVARYP	CRØSIE	BCNVRT	CØRØUT	TWRIT	CØUFCN
CRØSY	ISDARR	PRCØM	KNERGY	IRHIST	CØMBCR	RØSY1
ANTENA	RMAIN1	IFEMRS	CEFMRS	IMAIN1	RPØØL2	IPØØL1
XDBUG	CØRØSY	SIMDAT	RFIELD	CRATIØ	RATTDE	PWHEEL
MØMENT	ECNSTS	CSØLAR	CØNSTS	CFNDGB	CFNALP	CCNVRT
PLØT	PLØTF	AICRT3	ENERGY	IDFRMV	PLTND	

2. Segment B, Size 23 K

READER	READZ	READC	BLØCK4	CRINIT	SRINIT	ARFIC
READH						

3. Segment C, Size 230 K

ADCAL	ADMIMP	AIRDRG	ARANGE	ATTUDE	BØUNDS	CKINPT
SØLAR	CØMBNZ	DAMPER	DARCØS	DARSIN	DEREQ1	DSAVE
FASTR	FIELD	FINDGB	FINDS	FNDALP	FNDGAM	FNYBCM
NXNSØL	MYPMAT	MATMPY	MAGNTS	ISPIN	INVERT	HAG
ØMCL	PINTRP	PLH	RELØC	RELØC2	RTSIDE	SET
VISCUS	TPREAD	SUNLIN	SUNDEP	SUBCØN	SINPUT	SETVAL
WHEELS	XACM	XYZPLH	ZSPIN	GRAM	GINPUT	XFIND
GPRINT	GØUT	RØSY	CØNVRT	LSARAY	LDARAY	WRITEP
UFCN	WRITER	SØUT	CANTNA	DERFND	CØMALP	RPØØL7
RPØØL9	ØUTTWØ	ELMNTS	VARBLS	RPØØL8	CØMSØL	ØUTFØR
STAPØS	RPØØL4	RPØØL6	ØUTTHR			

4. Segment D, Size 106 K

ADM4RK	ASINR	ATANQR	CØMPEX	CØNVIN	DEREQ	FMERIS
ØUTPUT	ØRBEL	NØINTP	MAIN2	JULDAY	GETTAP	FINDX
READE	SUB3	SUB4	SUB5	TRANSP	ZERØ	AFMRIS
CØRBIT	IØRBIT	IØUTPT	ACØSR			

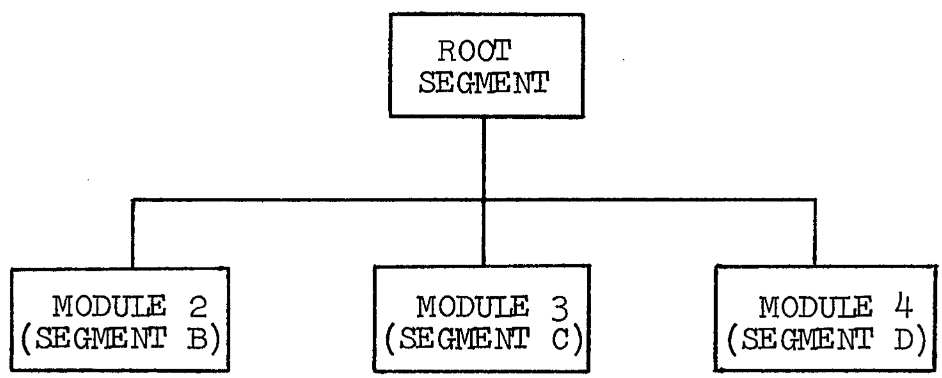


Figure 6 - OVERLAY TREE STRUCTURE

JOB CONTROL LANGUAGE LISTING

Column
1

```
//YVPLB001 JOB(P20101861F,P,NA0103,004005),077,MSGLEVEL=(1,1)
//LINK EXEC LINKGO,PARM.LINK='OVLY,LET,LIST,XREF,MAP',
//     REGION.GO=480K,COND=(382,EQ)
//LINK.TAPELIB DD DSN=PLB,VOL=SER=TD6052,LABEL=(1,BLP),
//     DISP=(OLD,DELETE)
//LINK.SYSLIN DD *
//*
//*  PUT AVCOPAC OBJECT DECK HERE
//*
 INCLUDE TAPELIB
 INSERT LA000000,MAIN,ADDAM,ACNVRT,ARFDT2,ARTLU1,CHECK,COMP62
 INSERT DROUT,ELOUT,FVAL,H1,HMSOUT,INOUT,MULTM,PLOT,PLOTF,QTIME
 INSERT STDPND,STVRBL,TCNVRT,YRMODA,AICRT3,IDFRMV,PLTND
 INSERT ADREAL,ANTENA,CCMBNZ,CCNVRT,CDMPER,CEFMRS,CFINDX
 INSERT CFNDGB,CFNALP,CODPLY,CONSTS,CRATIO,CSOLAR,CSTVAL,DATOUT
 INSERT DEPLOY, ECNSTS,EISUBK,GLITCH,HAMOUT,HEAD22,IDATE1
 INSERT IFASTR,IFEMRS,IMAIN1,IMAIN2,IODPLY,IPOOL1,ITEMP,ITEMP2
 INSERT JBTEST,MOMENT,NODER,OUTONE,PLOTIT,PRCOM,PWHEEL
 INSERT RATTDE,RDAMPR,RFASTR,RFIELD,RMAIN1,RMAIN2,RMGNTC
 INSERT RPOOL1,RPOOL2,RPOOL3,RPOOL5,RSUNCL,RTDIST,RVISCS
 INSERT SATPOS, SUNANG,SWITCH,XIN1,XIN2,XIN3,XIN4,ZSPINR
 INSERT NUM
 OVERLAY A1
 INSERT CRINIT,READH,SRINIT,READER,READZ,READC,BLOCK4,ARFIC
 OVERLAY A1
 INSERT ROSY,WRITEP,WRITER,GRAM,LDARAY,UFCN,LSARAY
 INSERT ADCAL,ADMIMP,AIRDRG,ARANGE,ATTUDE,BOUNDS,CKINPT
 INSERT COMBNZ,CONVRT,DAMPER,DARCOS,DARSIN,DEREQ1,DSAVE,FASTR
 INSERT FIELD,FINDGB,FINDS,FNDALP,FNDGAM,FNYBCM,GINPUT,GOUT
 INSERT GPRINT,HAG,INVERT,ISPIN,MAGNTS,MATMPY,MPYMAT
 INSERT NXNSOL,OMCL,PINTRP,PLH,RELOC,RELOC2,RTSIDE,SET
 INSERT SETVAL,SINPUT,SOLAR,SOUT,SUBCON,SUNDEP,SUNLIN,TPREAD
 INSERT VISCUS,WHEELS,XACM,XFIND,XYZPLH,ZSPIN
 INSERT COMALP,COMSOL,CANTNA,DERFND,ELMNTS,OUTTWO,OUTFOR
 INSERT OUTTHR,RPOOL4,RPOOL6,RPOOL7,RPOOL8,RPOOL9
 INSERT STAPOS,VARBLS
 OVERLAY A1
 INSERT AFMRIS,ACOSR,ADM4RK,ASINR,ATANQR,COMPEX,CONVIN,DEREQ
 INSERT FMERIS,FINDX,GETTAP,JULDAY,MAIN2,NOINTP,ORBEL,OUTPUT
 INSERT READE,SUB3,SUB4,SUB5,TRANSP,ZERO,CORBIT,IORBIT,IOUTPT
 ENTRY MAIN
```

JOB CONTROL LANGUAGE LISTING (Continued)

Column
1

```
//GO.FT10F001 DD UNIT=(9TRACK,,DEFER),LABEL=(1,BLP),
//       VOL=SER=TD5244,DCB=(DEN=2,RECFM=VS,LRECL=7460,BLKSIZE=7464),
//       DISP=(OLD,KEEP)
//*      TD5244 IS JPL EPHEMERIS TAPE
//GO.FT11F001 DD UNIT=(2400-4,,DEFER),LABEL=(1,BLP),
//       VOL=SER=POOL,DCB=(RECFM=VBS,LRECL=124,BLKSIZE=2484),
//       DISP=(OLD,PASS)
//GO.FT12F001 DD UNIT=2314,DISP=(NEW,DELETE),SPACE=(CYL,(2,2)),
//       DCB=(RECFM=VBS,LRECL=412,BLKSIZE=416)
//GO.FT15F001 DD LABEL=(1,BLP),UNIT=(9TRACK,,DEFER),
//       DISP=(NEW,KEEP),VOL=SER=TD6212,
//       DCB=(RECFM=VS,LRECL=796,BLKSIZE=800)
//*      TD6212 IS A FLIGHT DATA SET
//GO.FT21F001 DD DSN=RESTART,LABEL=(1,BLP),UNIT=(9TRACK,,DEFER),
//       VOL=SER=TD6401,DCB=(RECFM=VS,LRECL=796,BLKSIZE=7964),
//       DISP=(NEW,KEEP)
//*      TD6401 IS A RESTART TAPE
//GO.SYSABEND DD SYSOUT=A
//GO.DATA5 DD *
```

3.6 User's Instructions

The following pragraphs are intended to aid the user in the operation of the corrector option of this program. Whether the program is used in a simulation mode or a corrector mode, a complete set of data to define the satellite, orbit, initial conditions, and environment is required. All the control words for the corrector mode are preset so that corrector mode operation is not invoked. This means that a normal simulation mode run requires only the usual input.

3.6.1 Writing Flight Data Sets

Test flight data sets can be written with the program if desired. When writing a test flight data set, the program is operating in the simulation mode. The input control words NHISTS 1 and IFLITE 0 create standard flight data sets. The input control words NHISTS 0 and IFLITE 1 create general flight data sets. The input control words NHISTS and IFLITE are preset to zero. Since the corrector control words are preset to not invoke corrector operation, none of the corrector control words should be input to write a test flight data set.

3.6.2 Corrector Operation

There are three basic requirements for using the corrector. The flight data and simulator data must be compatible; the type of error function or performance criterion must be selected; and the parameters to be varied must be defined. A discussion of these aspects of corrector use is presented below.

Flight data is read from a data tape and stored in common in a particular form for use by the corrector. The reading of flight

data is accomplished in the subroutine READH. The current program can read tapes created in either of the standard forms or tapes created in the general form described in Section 3.3. However, it is likely that the reading of real data tapes will require a modification to READH. In making this modification the structure of the resulting data storage in common should be strictly maintained. The common storage of data has the following format. The array is in double precision. The first NDWØRD locations are used for the temporary storage of each data record as it is read from the tape. The subsequent locations in the array are used for sequential storage of the pertinent data from the data records read from the flight data tape. For each data record, the time in seconds from the start of the year is stored first. The data for the appropriate data channels is stored in the next IDCHAN locations. This format is repeated for each time point in the data span. The resulting common array is as follows:

A(1) ⋮ A(NDWØRD)	Temporary storage for data records from flight data tape
TSTART DATA(1) DATA(2) DATA(3) . DATA(IDCHAN)	Stored data at time TSTART
TIME DATA(1) DATA(2) DATA(3) . DATA(IDCHAN)	Stored data at intermediate time
TSTØP DATA(1) DATA(2) DATA(3) . DATA(IDCHAN)	Stored data at time TSTØP

In evaluating the performance criteria for the corrector, simulator data must be put into an array at each print time for the simulator. The simulator data must be put into the data channels in the same order as has been chosen for the flight data. This can be accomplished by the input described in Section 3.3. In the event that real flight data is not compatible with the normal data produced by the simulator, suitable adjustments in the program will be required.

The selection of a particular type of performance criterion has no clear cut guidelines and only experience with each particular situation can determine a "best" criterion. It is probably advisable to put all the data on some nondimensional basis. For instance, if a criterion is being made up of both attitude angles and element displacements, the element displacements should possibly be nondimensionalized by weighting with the reciprocal of the antenna length. The best procedure may be to try several different criterion to see if any improvement in optimization can be observed.

The mechanics of defining the parameters which are to be varied consists of specifying the appropriate parameter control words and initial variations as described in Section 3.3. The more difficult problem is to decide which parameters should be varied. The first step in fitting of a given data span is to optimize the initial state vector particularly with respect to rates. Once the initial conditions have been resolved, the investigation of physical parameters of the system can be undertaken. There are no clear rules for selection of parameters. Selection

is more basically related to an evaluation of what parameters can explain the observed performance and an estimate of the inherent uncertainty in the knowledge of a particular parameter. The careful combination of these two concepts can lead to an estimate of a "most likely" set of parameters.

3.6.3 General Instructions and Recommendations

The following comments are intended as guidelines in using the corrector option.

The internal orbit option should be used for all corrector operation.

Intermittent printing of simulator output for corrector trials should be used to conserve output (NPRIN 10 is reasonable).

Plotting of simulator output for corrector trials should not be used (IPLØT 0).

When using a restart tape, the complete data deck for the original run must be used.

When using a restart tape, the flight data tape is not required.

4.0 DETERMINATION OF ANTENNA POSITIONS FROM ATTITUDE DATA

An investigation was undertaken to evaluate the feasibility of determining definitive antenna positions on the basis of attitude data for a spacecraft of the RAE type. This study consisted of two distinct efforts. The first was an examination of a simplified linear formulation of the equations of motion. The second was the use of the corrector to demonstrate the problems of determining definitive antenna positions. These two areas are discussed in the following paragraphs.

4.1 Simplified Equations of Motion

The equations of motion for the RAE satellite are quite complex. For illustrative purposes it is desirable to consider a simpler set of equations of motion associated with an X boom configuration. The equations of motion are taken from Reference 4. The planar geometry of the situation is shown in Figure 7.

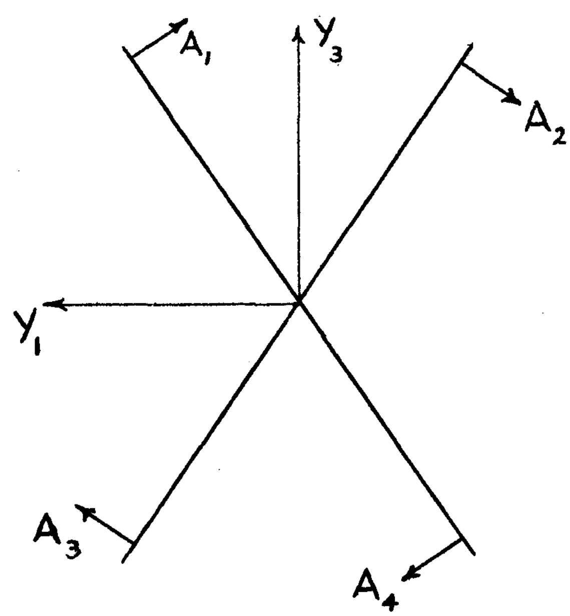


Figure 7 - Geometry of X Configuration

The equations of motion for pitch and in plane antenna displacements can be written in the following simplified form

pitch equation of motion (linearized)

$$I_2 \ddot{\beta} - M_{21} \ddot{q}_1 + C_{22} \dot{q}_2 + K_{21} q_1 + 3\Omega_0^2 (I_1 - I_3)\beta = M_2$$

element in plane equations of motion (linearized)

$$-M_{21} \ddot{\beta} + m_1 \ddot{q}_1 + k_1 q_1 = Q_1$$

$$m_2 \ddot{q}_2 + C_{22} \dot{\beta} + k_2 q_2 = Q_2$$

$$m_3 \ddot{q}_3 + k_3 q_3 = Q_3$$

$$m_4 \ddot{q}_4 + k_4 q_4 = Q_4$$

The element displacements associated with the generalized coordinates q_1, q_2, q_3, and q_4 are shown in Figure 8.

An examination of these simplified equations of motion provides the essential information for evaluating the feasibility of determining definitive antenna positions from attitude data. The motions associated with the generalized coordinates q_1 and q_2 are coupled to pitch but the motions associated with q_3 and q_4 have no effect on attitude in the linear approximation. Furthermore, the motion associated with q_2 is only coupled to attitude by the fact that the antenna is bent by gravity gradient forces. From these observations, it is clear that the feasibility of determining definitive antenna position from attitude data is questionable.

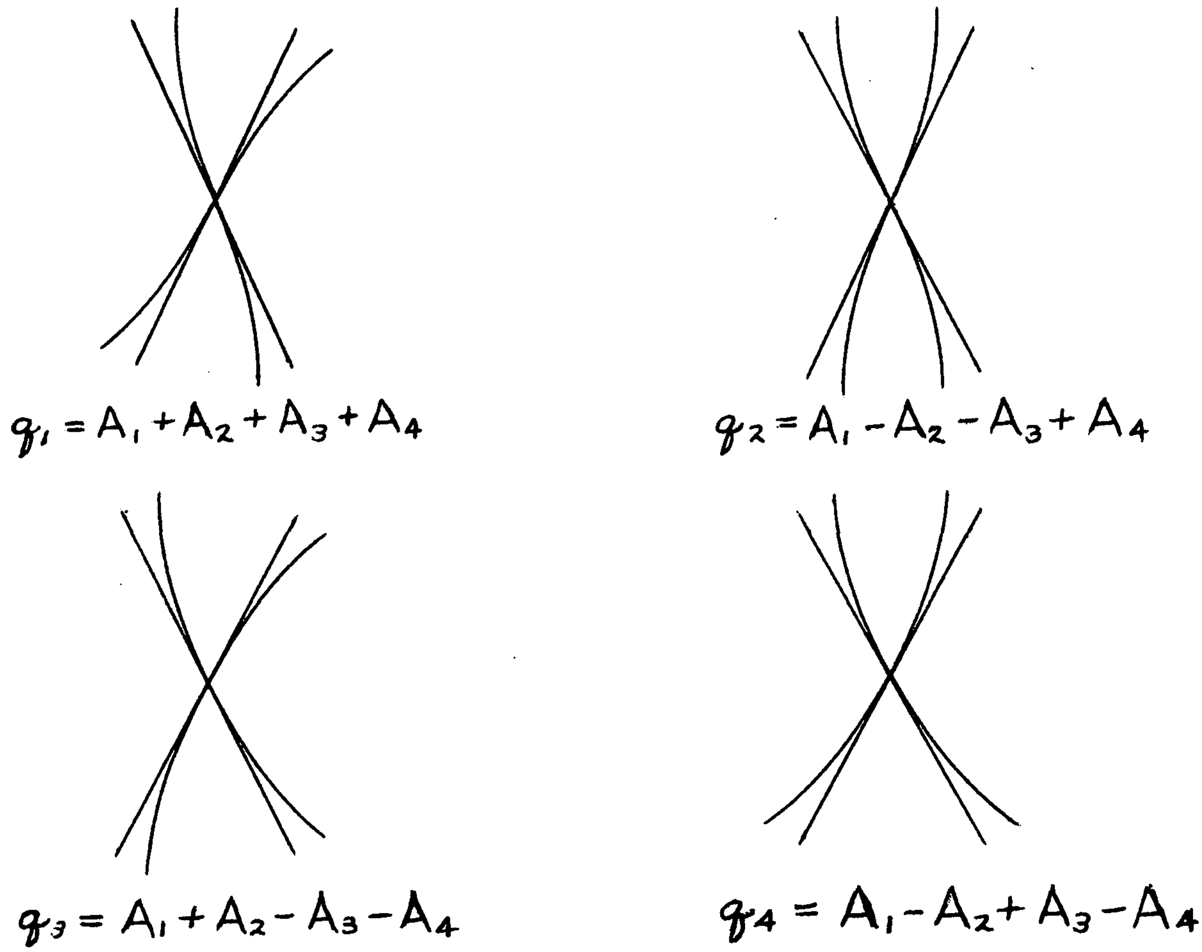


Figure 8 - Deformations for Generalized Coordinates

4.2 Illustrative Corrector Runs

In order to demonstrate the problems associated with determining definitive antenna positions from attitude data, two corrector runs were made. In these runs, the initial conditions on the in plane antenna displacements were treated as parameters. The flight data for the illustrative cases was created by the program. This means that, as far as the corrector operation was concerned, the data was perfect and the simulator model was perfect. Therefore, the results obtained represent the best estimates of antenna positions that can be expected for the limited number of runs performed.

The true flight data initial conditions on in plane displacements for the two cases were the following:

$A_1 = 140.53 \quad A_2 = -163.58 \quad A_3 = -162.92 \quad A_4 = 111.55$

or $q_1 = -74.42 \quad q_2 = 578.58 \quad q_3 = 27.98 \quad q_4 = 29.64$

The initial estimates of the in plane displacements for the two cases were the following:

$A_1 = 140.0 \quad A_2 = -140.0 \quad A_3 = -140.0 \quad A_4 = 140.0$

or $q_1 = 0.0 \quad q_2 = 560.0 \quad q_3 = 0.0 \quad q_4 = 0.0$

The estimates roughly correspond to gravity gradient equilibrium displacements.

The first corrector run considered a data span of sixteen minutes with points at thirty second intervals resulting in thirty-three data points for each data channel. The results of successful trials are given below for the first two stages. An unweighted squared error function was used.

Trial No.	A_1	A_2	A_3	A_4	Performance Criterion
3	137.5	-140.0	-140.0	140.0	63.055
4	132.5	-140.0	-140.0	140.0	54.688
5	122.5	-140.0	-140.0	140.0	39.774
6	102.5	-140.0	-140.0	140.0	17.204
7	62.5	-140.0	-140.0	140.0	0.996
9	62.5	-135.0	-140.0	140.0	0.951
12	62.5	-135.0	-142.4	140.0	0.836
17	62.5	-135.0	-142.4	139.4	0.835
End of First Stage $q_1 = -75.5$, $q_2 = 479.3$, $q_3 = -69.5$, $q_4 = -84.3$					
23	64.9	-135.0	-142.4	139.4	0.829
26	64.8	-136.3	-141.8	139.5	0.833
29	64.8	-136.6	-142.4	139.4	0.809
31	64.8	-136.6	-142.5	139.7	0.813
End of Second Stage $q_1 = -74.8$, $q_2 = 483.2$, $q_3 = -68.8$, $q_4 = -84.3$					

2

The standard deviation of attitude error for the best performance

$$\sigma = \sqrt{\frac{0.809}{3(33)}} = 0.090$$

The second corrector run considered a data span of sixty minutes with points at thirty second intervals resulting in one hundred and twenty-one data points for each data channel. The results of successful trials are given below for the first two stages. An unweighted squared error function was used.

Trial No.	A_1	A_2	A_3	A_4	Performance Criterion
3	137.5	-140.0	-140.0	140.0	169.038
4	132.5	-140.0	-140.0	140.0	146.388
5	122.5	-140.0	-140.0	140.0	106.312
6	102.5	-140.0	-140.0	140.0	47.631
7	62.5	-140.0	-140.0	140.0	11.225
11	62.5	-138.8	-140.0	140.0	11.227
14	62.5	-138.8	-142.5	140.0	11.215
16	62.5	-138.8	-142.5	145.0	11.192
End of First Stage $q_1 = -73.8$, $q_2 = 488.8$, $q_3 = -78.8$, $q_4 = -86.2$					
22	64.9	-138.8	-142.4	144.8	11.079
24	65.0	-138.7	-142.7	145.4	11.180
27	64.8	-139.3	-142.8	145.5	10.995
28	64.4	-140.4	-142.9	145.8	10.715
29	63.6	-142.7	-143.1	146.3	10.489
32	63.5	-142.9	-143.2	145.1	10.506
End of Second Stage $q_1 = -75.9$, $q_2 = 495.7$, $q_3 = -82.3$, $q_4 = -83.1$					

The standard deviation of attitude error for the best performance is given by:

$$\sigma = \sqrt{\frac{10.489}{3(121)}} = 0.17$$

The results of these two corrector cases illustrates the difficulties inherent in trying to determine definitive antenna positions from attitude data. However, some observations on these

results can be made. The procedure can quickly determine the system generalized coordinate q_1. The coupling of the coordinate q_2 is quite weak and the study of longer data spans would be required to resolve the question of determining q_2. The generalized coordinates q_3 and q_4 can probably not be resolved directly by use of attitude data alone. Finally, in using the result of these runs the best estimate of the true initial conditions would be to add the displacement pattern associated with q_1 to a symmetric equilibrium pattern for q_2. This would yield the following set of estimates for the initial in plane displacements:

$A_1 = 122.0 \qquad A_2 = -158.0 \qquad A_3 = -158.0 \qquad A_4 = 122.0$

4.3 Conclusions

The investigation into the feasibility of determining definitive antenna positions for the RAE configuration from attitude data alone leads to the following conclusions:

1) It is not, in general, possible to determine definitive antenna positions from attitude data alone.
2) The deformations associated with the generalized coordinate q_1 (Figure 8) are readily determined from attitude data.
3) It may be possible to determine motions of the generalized coordinate q_2 (Figure 8) if they are large enough and good quality data exists for long periods of time (several orbits).

4) Small motions associated with the generalized coordinates q_3 and q_4 (Figure 8) cannot be determined from attitude data.

5.0 REFERENCES

1. "User's Manual for RAE In-Orbit Simulator Computer Program," AVSSD-0017-69-CR, Contract No. NAS5-11050, Prepared by AVCO Government Products Group, Space Systems Division, Lowell, Massachusetts, January 1969.

2. "User's Manual for IMP Dynamics Computer Program," AVSD-0191-71-CR, Contract No. NAS5-11149, Prepared by AVCO Government Products Group, Systems Division, Wilmington, Massachusetts, March 1971.

3. Rosenbrock, H. H., "An Automatic Method for Finding the Greatest or Least Value of a Function," The Computer Journal, Volume 3, October 1960, pp. 175-184.

4. Final Report for A Study of Certain Dynamics Problems of the Radio Astronomy Explorer-Lunar (RAE-B) Satellite, AVSD-0060-71-RR, Contract No. NAS5-11801 MOD 4, Prepared by AVCO Corporation, Systems Division, Wilmington, Massachusetts, February 14, 1971.